Nonetheless, we will still use a distance vector approach may be used for updating the BF Table, except that some modification as described below is needed.

In a conventional distance vector update protocol, each node initializes its distance vector with distances to all its neighbors. It then sends the distance vector to all its neighbors. When a neighbor receives the distance vector, the neighbor updates its own distance vector if any shorter path is found. This neighbor then sends its update to all its own neighbors. The procedure keeps on going until the algorithm converges.

As there are old and STAR bridges in the bridged LAN, the conventional distance vector update protocol cannot be applied directly. Each bridge knows only the cost to its direct neighbors. If x and y are distant STAR neighbors, both x and y do not know $d_T(x, y)$ since there are one or more old bridges between them. Therefore, a bridge $x \in B$ has to determine $d_T(x, y)$ if y is a distant STAR neighbor. Unfortunately, due to the limitation of old bridges, STAR bridges may not be able to determine every distance correctly. Nonetheless, the distances may be estimated such that each estimated distance is at least its corresponding real distance.

The process for path finding in the proposed protocol consists of two procedures:

- Distance Vector Estimation
- Distance Vector Enhancement

In these procedures, STAR bridges communicate using Distance Vector Computation SBPDUs. In the Distance Vector Estimation procedure, each STAR bridge initializes the distance to its neighbors in the STAR bridge graph of the bridged LAN. It involves discovery of distant STAR neighbors and computation of the tree path distances as described above. In the Distance Vector Enhancement procedure, which follows the Distance Vector Estimation procedure, STAR bridges exchange their distance vectors, discover other non-neighbor STAR bridges, and find the shortest path to them.

In the process, a STAR bridge n maintains a distance vector only for other STAR bridges known to n. As unknown STAR bridges are discovered by n in the process of the algorithm, a new entry is created in the distance vector maintained by n for each newly discovered STAR bridge. When the process ends, n should have discovered all other STAR bridges and its distance vector will consist of one entry for each STAR bridge $n' \in B\backslash\{n\}$. Each entry in the distance vector of n consists of a tuple of seven fields. The entry associated with n' in the distance vector of n is denoted as $DVT(n, n')$. As summarized in Table 2, the information contained in $DVT(n, n')$ provides an estimated distance between n and n', indicates whether the estimated distance is actually accurate, and enables STAR bridge n to know its forwarding port for n', its next hop STAR bridge neighbor on the forwarding path to n', indicates whether the forwarding path is a tree path, as well as whether n' is an ancestor or a descendant. Incidentally,

$d(n, n')$, the current estimated distance from n to n', will be appropriately initialized as described in Sections III.A and III.B.

Field	Definition
N'	ID of destination STAR bridge
$d(n, n')$	Estimated distance between n and n'
$F(n, n')$	Forwarding port for n'
$next(n, n')$	ID of the next hop STAR bridge neighbor on the path from n to n'
$FG\_A(n, n')$	Distance accuracy flag with a value 1 if $d(n, n')$ is accurate, and 0 otherwise
$FG\_T(n, n')$	Tree path flag with a value 1 if the path from n to n' is a tree path, and 0 otherwise
$FG\_R(n, n')$	Relation flag with a value 1 if n' is an ancestor of n, -1 if n' is a descendant of n, and 0 otherwise

Table 2: Fields in *DVT(n, n')* for a Path from *n* to *n'*

III.A Distance Vector Estimation

In the DV Estimation procedure, a STAR bridge n discovers all its STAR neighbors, both direct and distant, that is, the fields in the distance vector of STAR bridge n are filled out for each STAR bridge $n' \in B\{n\}$ where $(n, n') \in E_B$. There are two phases in the Distance Vector Estimation procedure. In the first phase, STAR bridge n estimates $d_T(n, n')$ if n' is a distant STAR neighbor and n fills out the entry for n' in $DVT(n, n')$. Note that $d(n, n')$ is equal to the estimated $d_T(n, n')$ in this phase. In the second phase, n determines and fills out the entry for k if $k \in N_B(n)$. If k is a distant STAR neighbor of n, then n replaces $d(n, k)$ by $c(n, k)$ and other fields accordingly only if it is appropriate. This phase will be discussed in more detail later.

Before the spanning tree algorithm starts, each bridge, either old or STAR, should know its own ID and the cost of the link to each of its direct neighbors in the bridged LAN. After the tree has been built, every bridge k will also know its tree links, as well as the root bridge and the root path distance, $d_r(k)$, where bridge r is the root bridge. Incidentally, $d_r(r) = 0$. Table 3 is the topology information of bridge v in FIG. 10 after the spanning tree computation. The column Old/STAR is the information obtained by STAR bridges only while all bridges, either old or STAR, obtain all other columns.

Bridge	Type	Distance	P(v)	Old/STAR
R	Root	$d_r(v)$	$p_r(v)$	N/A
X	Parent	$c(v, x)$	$p(v, x)$	Old
U'	Non-tree neighbor	$c(v, u')$	$p(v, u')$	STAR

Table 3: Topology Database of Bridge *v*

There are three kinds of DVC_SBPDU frames in the first phase – *DVMyInfo*, *DVOurInfo*, and *DVInform* frames. Table 4 shows the formats of these frames. DVMyInfo frames are used for a STAR bridge to inform other STAR bridges of its own topology information. DVOurInfo frames carry information related to both the source and the destination STAR bridges. DVInform frames allow STAR bridges to pass on topology information of other STAR bridges. DVRecord frames are used in Distance Vector Enhancement and will be described in Section III.B DVC_SBPDU_Proc procedure first identifies the frame and then invokes corresponding procedures. The flow-chart for DVC_SBPDU_Proc is shown in FIG.12.

Each STAR bridge k sends a DVMyInfo(k) frame on its root link only if its parent is an old bridge. If there is any STAR bridge along the root path of k, the one that is nearest to k, say n, will receive the DVMyInfo(k) frame from a child link. Note that n and k are distant STAR neighbors. n is the distant STAR ancestor neighbor of k and $n = dsan(k)$. Bridge n can determine the tree path distance between k and n, wherein $d_T(n, k) = d_T(k, n)$ is the difference of their root path distances. Then, n informs k of the distance between them by a DVOurInfo(n, k) frame and stops forwarding the DVMyInfo(k) frame. In this case, n and k are on the same branch and n is an ancestor of k. If the STAR bridges are on different branches, like v and v' in FIG.10, and there is no STAR bridge on the tree path between them, then, since DVMyInfo frames are multicast frames addressed to all STAR bridges, v' will receive the DVMyInfo(v) frame of v and vice versa. However, there is no way for them to calculate the real tree path distance between them using root path distance alone. In the case of v and v', if they know that they have the same parent, they can determine the accurate tree path distance. Therefore, the DVMyInfo(v) frame also contains the information of the parent of v. When v and v' receive each other's DVMyInfo frame through a root link and find out they are siblings, they may calculate the distance between them correctly by adding $c(v, parent(v))$ and $c(v', parent(v'))$.